

LIBRARY OF THE
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

510.84

I26r

no.505.510

cop.2



CENTRAL CIRCULATION BOOKSTACKS

The person charging this material is responsible for its renewal or its return to the library from which it was borrowed on or before the **Latest Date** stamped below. **The Minimum Fee for each Lost Book is \$50.00.**

Theft, mutilation, and underlining of books are reasons for disciplinary action and may result in dismissal from the University.

TO RENEW CALL TELEPHONE CENTER, 333-8400

UNIVERSITY OF ILLINOIS LIBRARY AT URBANA-CHAMPAIGN

APR 04 1994

MAR 2 / 1994

MAR 27 1993

JAN 06 1998

When renewing by phone, write new due date below
previous due date.

L162

510.84
IL62
no. 505
cop. 2

Report No. UIUCDCS-R72-505

NETWORK COMPUTER ANALYSIS

by

Edward K. Bowdon, Sr.



DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

THE UNIVERSITY OF ILLINOIS
MAY 20 1972
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN



Digitized by the Internet Archive
in 2013

<http://archive.org/details/networkcomputera505bowd>

Report No. UIUCDCS-R72-505

NETWORK COMPUTER ANALYSIS

by

Edward K. Bowdon, Sr.

Computer Science Department
University of Illinois
Urbana, Illinois 61801

(This research was supported in part by the National Science Foundation
under Grant No. NSF GJ 28289.)

5/10/84
ILen
24. 505-510
Page 2

ABSTRACT

This research is aimed at developing analytical tools for system modeling and analysis of real time computer networks. We begin by formulating a geographically distributed network of computers.

From a postulation of the essential characteristics of our computer network, we develop a queueing theory model for a multiserver system with a finite length priority queue. Then, under the assumptions of Poisson input and exponentially distributed processing times, we utilize this idealized mathematical model to investigate the steady state stochastic behavior of jobs in the network. We are particularly interested in the efficiency of computer utilization and the average waiting time for jobs of different priority classes.

We conclude by employing the average waiting time results from the queueing theory model to determine steady state job dispatching rules for a network of centers. From a knowledge of the average arrival rates and processing times for each priority class, we determine what fraction of the jobs in each class should be transmitted between centers in order to balance the average waiting times for each priority class throughout the network.

1. INTRODUCTION

A historical survey of computer system development reveals that the application demands and requirements of the user have greatly influenced the design philosophy of computer systems. The demand for a continuously improving cost/performance index and the requirements imposed by an ever growing number of esoteric computer applications has antiquated the current design philosophy. In order to improve the cost/performance index and still maintain generality, the current design approach requires larger and faster central processors (including main memories) and necessitates the development of sophisticated software systems and programming techniques.

Meanwhile, peripheral devices have remained comparatively primitive, their performance requiring substantial software support from the system. As a consequence, there exists a considerable cost imbalance between the hardware and software. Recent technological advances in the large scale integration of electronic circuits now make it feasible to restore this economic balance by decentralizing the central processor's tasks and, at the same time, simplifying the software system. We envisage a special purpose hardware/firmware processor as a substitute for the program routines (or even entire algorithmic functions) that have been ordered structures and are frequently used in a large spectrum of applications.

Interconnection of these processors in a geographically distributed, multiple access, network computer would provide computing efficiency through the sharing of hardware, software, and personnel. If a remote user can be serviced in a manner that is indistinguishable from the service rendered at a computer installation, he can realize the full power of a large-scale

computer complex at costs that can make computer ownership impractical and/or uneconomical. Additionally, the software control of the network is greatly simplified, which in turn results in an increase in the system throughput.

To those with mutual interests, the network affords an effective means to communicate remotely and dynamically integrate dispersed activities. The network's ability to shorten significantly the turn around time allows the user to react to changing problem factors quickly and thereby produce cost savings.

Such a geographically distributed computer network is a preeminent candidate for future generation computer systems and, therefore, the object of our research.

2. ANATOMY OF A NETWORK

Structure

In our analysis of networks computers, we are interested in three distinct types of networks. Figure 1 illustrates a regional network of computing centers interconnected by common carrier facilities.



Figure 1. Nationwide computing center network.

Figure 2 illustrates a time shared computer center consisting of one or more computers connected by a high speed exchange loop.

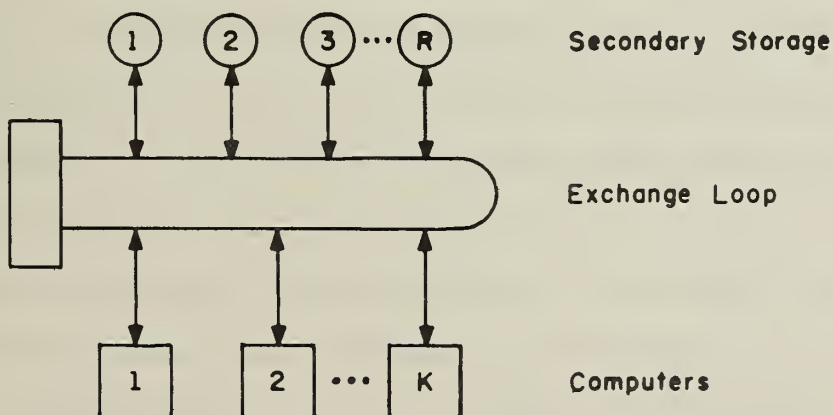


Figure 2. A typical computing center.

Figure 3 illustrates a tree network for a store and forward communications facility.

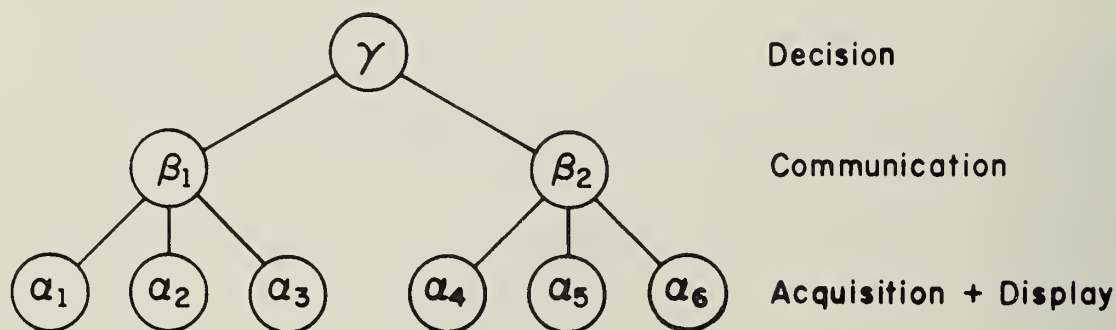


Figure 3. A three level communications network.

Owing to the large size of many organizations and to the dispersion of their facilities, we are interested in a network computer consisting of geographically distributed computing centers. Each computing center contains a number of computers, bulk storage devices, and peripheral devices interconnected by a high-speed communications facility called an exchange loop. This exchange loop is divided into a number of independently accessible time division channels, thus providing switchable interconnectability and intercommunications between all computers and devices in a center.

The computing centers, located at each node of the network, are interconnected by medium speed common carrier facilities. The capacity of each of these trunks may be considerably lower than the bit rate of facilities within the center, since each center's communities of interest usually require only occasional data transfers between centers. As a result, a major amount of processing occurs within each center with only a limited amount of intercenter communications.

Viewing each computing center as a node of a communications network, the system has the overall appearance of one large processing facility which uses common carrier facilities as transfer links between geographically separated nodes. The structure of this network may be likened to one large processing "machine." This "machine" is organized into a hierarchy of structural levels as follows:

1. Network
2. Centers
3. Computers

Each of these levels may in turn be viewed as a computing "machine" which performs some fraction of the total work. Under the control of an operating system, this structure allows independent processing jobs to be executed simultaneously within a plurality of computers and, further, permits common access by these computers to the entire mass storage system.

A key feature in the network is the concept of a common file system with uniform accessibility to all files by all computers. This could be accomplished by assigning each disk file to a computer and then, allowing only that computer to write on the disk file, even though any computer may read from it. This would mean that a computer never writes into any space for which it does not have direct cognizance and responsibility. As a result, the problems of space allocation and release become relatively simple; contention would never exist for the use of file space.

Implicit in this common file system is the ability to effect the movement of data files from one computer to another within a center without physically moving the data files. For example, when a program segment is executed and an output record is produced, this output record would be written on the disk file assigned to this computer. In order to "move" this

data file to another computer it would only be necessary to send the address of the data file in secondary storage to the intended computer. By virtue of the exchange loop, all disk files are equally accessible to all computers, and thus the intended computer merely would use the address of the data file to access the intended data. Notice that this mechanism would always allow a program to write its output records on the disk file associated with the computer in which the program is running with no regard to potential users of these records.

Intercenter Communications

As noted earlier, the network of centers is interconnected by common carrier links. In order to maximize the percentage of time that the intercenter links are occupied, one computer in each center, designated the "communications computer," is assigned the responsibility of servicing all intercenter communications for that center.

This communications computer would contain a communications program which overlaps the transmission of records with the turn around time required for answerback responses from the far end of the link. The use of answerback responses and the retransmission of erroneous records on the link would minimize the amount of redundant information required to provide the desired level of error protection.

Each of the computers within a center would inform the communications computer of files to be transmitted or received. The communications computer, in turn, would inform the requesting computers of the receipt of requested files from other centers.

Operating System

The control mechanism for the network is provided by an operating system, resident in each computer, the primary function of which is to disseminate and regulate automatically the processing of jobs. When a job enters the system it may be assigned to any one of the computers in a center. The operating system provides access to the full parallel processing power of the network by detecting when work is ready to be performed and queueing this available work among all the computers in the center. This distributed operating system tends to minimize the degradation in system performance due to systems programs. However, since we are seeking optimum performance, the overhead inherent in the operating system, though small, must still be considered in the analysis.

Summary

The four major characteristics which make our network computer distinctive are that:

1. It is a multicomputer structure which is not limited to a particular configuration, but rather, is variable in nature and may be adapted to the requirements of a particular application. The exchange loop serves as an interconnection and switching matrix between all computers and devices in a center.
2. It is a geographically distributed network consisting of centers interconnected by common carrier links. These links terminate in each center into a computer containing communications programs capable of efficiently moving data between centers spaced hundreds of miles apart.

3. It is a symmetrical structure in which every address on the exchange loop is equally accessible from every computer on the loop. Common data files are stored in secondary storage devices and are readily accessible by all computers in the network.
4. It is a controlled structure which contains an operating system that is distributed among the computers in a center and which regulates the allocation of facilities and the assignment of work to the computers in the network.

These characteristics allow parallel processing among all computing facilities within the network without the need to preplan the exact distribution of work in terms of the various applied problems presented to the network.

3. RESEARCH METHODOLOGY

Suppose we have a network of computing centers each consisting of one or more computers with potentially different processing rates. When any given job enters a center it will be queued for processing. The exact manner in which jobs are queued will depend on the system structure, the availability of facilities, and the control procedures which govern the system.

Computer Network Modeling

We begin our development by formulating an idealized mathematical model for a computing center. Our motivation for using an idealized model is simply that of mathematical ease and tractability; at the same time, however, we must ensure that the idealizations introduced lead to a system model which retains the essential characteristics of the real system. In order to apply queueing theory we must first characterize the following:

1. The population from which jobs are drawn.
2. The arrival pattern of jobs.
3. The queue discipline.
4. The number and arrangement of computers.
5. The service pattern which governs the processing of jobs.

Our model may be described using the following terminology.

The input to the system is an infinite population of units having r priority classes (the smaller the index the greater the priority). Units of priority k , ($1 \leq k \leq r$), arrive as a Poisson process with interarrival time distribution $A_k(t)$, given by:

$$A_k(t) = 1 - e^{-\lambda_k t} \quad (t > 0)$$

and zero otherwise. Units from different priority classes arrive independently with mean arrival rates $\lambda_1, \lambda_2, \dots, \lambda_r$. The arrival rates are independent of the length of the queue.

The queue discipline is nonpreemptive priority. All jobs are considered patient in that once queued they remain in queue until processed. As a result, no jockeying, reneging, or balking is permitted. When a channel finishes servicing a unit, the channel is assigned to the highest priority unit present on queue, if any. Units of equal priority are serviced on a first come, first served basis. If a channel finishes servicing a unit and the queue is empty, the channel remains free until the next unit enters the system.

Our system [1] differs from previous priority models [7, 8, 15, 27, 53] in that the maximum queue length, M , is finite rather than being infinite. We note that when there are M units of priority k or higher in the queue, an arrival of priority k cannot join the queue since there is no room for it. This occurs with nonzero probability for finite queues. When this occurs we say that the source for units of priority k is blocked and no more units of priority k can arrive until there are less than M units of priority k or higher in the queue. We further assume that if a unit of priority k arrives at a time when the queue length is M , then it joins the queue only if a unit of priority less than k is among the M . In this case, the unit of lowest priority less than k is displaced and lost to the system.

The service mechanism consists of c identical channels in parallel which serve a common queue of maximum length M . The service times of each channel are exponentially distributed with density function, $b(x)$, given by:

$$b(x) = \mu e^{-\mu x} \quad (x \geq 0)$$

and zero otherwise. The mean service rate for a single channel is μ . The combined service rate for the system is given by:

$$\mu_c = \mu \min(c, n) \quad (0 \leq n \leq N)$$

where n is the number of units in the system and $N = c + M$. The service rate is independent of the length of the queue.

In sum, we are going to develop a queueing theory model for a nonpre-emptive priority system having r classes of Poisson input, c identical exponential service channels, and a finite queue of maximum length M .

For brevity, we define:

$$\sigma_k = \sum_{\ell=1}^k \lambda_{\ell} \quad \text{and} \quad \rho_k = \frac{\sigma_k}{c\mu} \quad (1 \leq k \leq r)$$

$$\lambda = \sigma_r \quad \text{and} \quad \rho = \rho_r$$

In what follows, unless otherwise noted, we shall tacitly assume $M \geq 0$, $\rho_k \neq 1$, and $1 \leq k \leq r$.

Temporal Behavior

Let $P_n(t)$ denote the probability that an arrival at time t finds n units in the system ($0 \leq n \leq N$). Let $P_{m,k}(t)$ denote the probability that an arrival of priority k at time t cannot be served immediately and will be m^{th} away from service; that is, he finds $(m-1)$ waiting units of priority k or higher, ($1 \leq m \leq M+1$). Also, let $p(>0)$ denote the steady state probability that an arrival of any priority does not go directly into service, that is, it waits in the queue. Then $p_{M+1,k}$ is the steady state probability of finding the queue full of priority k or higher units. Now the set of steady state probabilities $[p_n]$

totally characterizes the system, and at the same time the set of steady state probabilities $[1-p(>0), p_{m,k}]$ also totally characterizes the system. It will be convenient to have both sets.

The basic differential difference equations for our system have the form

$$P'_0(t) = -\lambda P_0(t) + \mu P_1(t) \quad (n=0) \quad (1)$$

$$P'_n(t) = -(\lambda + n\mu) P_n(t) + (n+1)\mu P_{n+1}(t) + \lambda P_{n-1}(t) \quad (0 < n < c) \quad (2)$$

$$P'_n(t) = -(\lambda + c\mu) P_n(t) + c\mu P_{n+1}(t) + \lambda P_{n-1}(t) \quad (c \leq n < N) \quad (3)$$

$$P'_N(t) = -c\mu P_N(t) + \lambda P_{N-1}(t) \quad (n=N) \quad (4)$$

For $M > 0$

$$P'_{1,k}(t) = -\sigma_k P_{1,k}(t) + \lambda P_{c-1}(t) + c\mu P_{2,k}(t) - c\mu P_c(t) \quad (m=1) \quad (5)$$

$$P'_{m,k}(t) = -(\sigma_k + c\mu) P_{m,k}(t) + c\mu P_{m+1,k}(t) + \sigma_k P_{m-1,k}(t) \quad (1 < m \leq M) \quad (6)$$

$$P'_{M+1,k}(t) = -c\mu P_{M+1,k}(t) + \sigma_k P_{M,k}(t) \quad (m=M+1) \quad (7)$$

For $M=0$

$$P'_{1,k}(t) = -c\mu P_{1,k}(t) + \lambda P_{c-1}(t) \quad (m=1) \quad (8)$$

Equations (1) through (3) are recognized as the well known differential equations for infinite length queues [27, 53]. The finite length queue imposes a boundary condition on the system which results in equation (4). Equation (5) results from the following argument. For $M > 0$, an arrival of priority k at time $t + \Delta t$

will be first away from service if and only if one of the following mutually exclusive events occur:

- (a) an arrival of priority k at time t would have been first in queue and in the interval Δt there were no arrivals of priority k or higher and no service.
- (b) an arrival of priority k at time t would have been second in queue and in the interval Δt there were no arrivals of priority k or higher but there was a service.
- (c) any arrival at time t would have found c-1 units in service and in the interval Δt there was an arrival and no service.
- (d) an arrival of priority k at time t would have been first in the queue (except for the case where any arrival would have found c units in service) and in the interval Δt there were no arrivals of priority k or higher but there was a service.

Now, within terms of Δt^2

$$P_{1,k}(t+\Delta t) = P_{1,k}(t) \{1 - \sigma_k \Delta t - c\mu \Delta t\} + P_{2,k}(t) \{c\mu \Delta t\} + P_{c-1}(t) \{\lambda \Delta t\} \\ + \{P_{1,k}(t) - P_c(t)\} \{c\mu \Delta t\}$$

Transposing the term $P_{1,k}(t)$, dividing the equation by Δt , and letting $\Delta t \rightarrow 0$, we obtain equation (5). Similar arguments yield equations (6-8).

Steady State Behavior

We readily obtain the steady state (time independent) difference equations by setting the left side of the differential equations to zero and solving for the higher order terms in n and m respectively. Solving the steady state difference equations recursively yields the following steady state probabilities:

$$p_n = p_0 (\lambda/\mu)^n / n! \quad (0 \leq n \leq c) \quad (9)$$

$$p_n = p_0 (\lambda/\mu)^n / (c! c^{n-c}) \quad (c \leq n \leq N) \quad (10)$$

where

$$p_0 = \left\{ \sum_{n=0}^{c-1} \frac{(cp)^c}{n!} + \frac{(cp)^c}{c!(1-\rho)} (1-\rho^{M+1}) \right\}^{-1} \quad (11)$$

$$p(>0) = p_0 \frac{(cp)^c}{c!(1-\rho)} (1-\rho^{M+1}) \quad (12)$$

$$p_{m,k} = p(>0) \rho^{m-1} (1-\rho_k) / (1-\rho_k^{M+1}) \quad (1 \leq m \leq M+1) \quad (13)$$

In particular we obtain equation (13) as follows. After recursively solving the steady state difference equations resulting from equations (5) through (8), we obtain:

$$p_{m,k} = \rho_k^{m-1} p_{1,k} \quad (1 \leq m \leq M+1)$$

Then

$$[1-p(>0)] + \sum_{m=1}^{M+1} p_{m,k} = 1$$

or

$$p(>0) = p_{1,k} \sum_{m=1}^{M+1} \rho_k^{m-1} = p_{1,k} (1-\rho_k^{M+1}) / (1-\rho_k)$$

from which

$$p_{1,k} = p(>0) (1-\rho_k) / (1-\rho_k^{M+1})$$

and equation (13) ensues. Then from equations (9) through (13) we derive the following system descriptors.

The average number of units of priority k or higher in the queue:

$$L_q(k) = \sum_{m=0}^M m p_{m+1,k} = \left\{ \frac{\rho_k}{1-\rho_k} - (M+1) \frac{\rho_k^{M+1}}{1-\rho_k^{M+1}} \right\} p(>0) \quad (14)$$

The average number of units waiting in the queue:

$$L_q = L_q(r) = \left\{ \frac{\rho}{1-\rho} - (M+1) \frac{\rho^{M+1}}{1-\rho^{M+1}} \right\} p(>0) \quad (15)$$

The average number of units of priority k waiting in the queue, ($1 < k \leq r$):

$$L_{qk} = L_q(k) - L_q(k-1) = \left\{ \frac{1}{1-\rho_k} - \frac{1}{1-\rho_{k-1}} - (M+1) \left(\frac{1}{1-\rho_k^{M+1}} - \frac{1}{1-\rho_{k-1}^{M+1}} \right) \right\} p(>0) \quad (16)$$

The average number of units in service:

$$L_s = \sum_{n=0}^{c-1} n P_n + c p(>0) = c \rho \{1 - p_0 \rho^M (c\rho)^c / c!\} \quad (17)$$

The mean number of free service channels:

$$c_f = c - L_s = c - c \rho \{1 - p_0 \rho^M (c\rho)^c / c!\} \quad (18)$$

The average number of units in the system both waiting and in service:

$$L = L_s + L_q = c \rho \{1 - p_0 \rho^M (c\rho)^c / c!\} + \left\{ \frac{\rho}{1-\rho} - (M+1) \frac{\rho^{M+1}}{1-\rho^{M+1}} \right\} p(>0) \quad (19)$$

Average Waiting Times

As we noted earlier, a unit of priority k will be displaced from the queue if an arrival of priority $(k-1)$ or higher finds M units of priority k or higher in the queue. If a unit of priority k , ($1 < k \leq r$), is displaced from the queue, theoretically, it is lost to the system and has an infinite waiting time.

Since this occurs with probability $[p_{M+1,k} - p_{M+1,k-1}]$, which is nonzero for

finite queues, the expected value of the waiting time for all units of priority less than $k=1$ is infinite. A more meaningful descriptor is the average waiting time in queue for units of priority k , which join the queue and are not displaced, denoted simply as W_k . We let $p(0 < W_k < \infty)$ denote the probability that a unit of priority k waits, but is not displaced by a higher priority unit and lost to the system. Since a priority 1 unit cannot be displaced, $p(0 < W_1 < \infty) = p(>0)$. But,

$$p(0 < W_k < \infty) = p(>0) - [p_{M+1,k} - p_{M+1,k-1}]$$

$$= \frac{1-\rho_k^M}{1-\rho_k^{M+1}} + \rho_{k-1}^M \frac{1-\rho_{k-1}^{M+1}}{1-\rho_{k-1}^{M+1}} p(>0) \quad (1 < k \leq r) \quad (20)$$

Given that a priority k unit, say T_k , waits in queue, there will be, on the average, n_k units of priority k or higher ahead of T_k . We expect to find $Lq(k)$ units of priority k or higher in the queue, but given that T_k waits we expect to find $n_k = [Lq(k) - M \cdot p_{M+1,k}] / p(>0)$ ahead of him. (The term $M \cdot p_{M+1,k}$ compensates for the situation in which a priority k unit finds the queue full and cannot join.)

The average waiting time in queue for units of priority 1, W_1 , is determined by the following argument. Given that a priority 1 unit, say T_1 , waits in queue, there will be on the average, n_1 units of priority 1 already in queue ahead of T_1 . After these n_1 units are served T_1 will be first in queue and must wait for one more unit to be served. Since the combined service rate when there is a queue is $c\mu$, T_1 waits on the average $(n_1+1)/c\mu$, given that he waits. Removing the condition on waiting and substituting for n_1 we obtain:

$$W_1 = \frac{1}{c\mu} \left\{ \frac{1}{1-\rho_1} - (M+1) \frac{\rho_1^M}{1-\rho_1^{M+1}} \right\} p(>0) \quad (21)$$

The waiting time in queue for units of priority k , ($1 < k \leq r$), is determined by the following argument. Given that a priority k unit, say T_k , waits in queue but is not displaced, there will be on the average n_k units of priority k or higher already in queue ahead of T_k . T_k will have to wait on the average, at least $(n_k+1)/c\mu$, given that he waits in queue but is not displaced. Removing the condition on waiting and not being displaced, we have:

$$W_k > p(0 < W_k < \infty) \cdot (n_k+1)/c\mu$$

Now while T_k is waiting in queue, units of priority $(k-1)$ or higher arrive and take a position in queue ahead of T_k . So, T_k must also wait for these units to be served. Since the higher priority units arrive at a rate σ_{k-1} for the duration of his waiting time, W_k , T_k must wait an additional amount $\sigma_{k-1} \cdot W_k / c\mu$. Hence

$$W_k = p(0 < W_k < \infty) \cdot (n_k+1)/c\mu + \sigma_{k-1} \cdot W_k / c\mu$$

After making use of the definition of ρ_{k-1} , transposing the term in W_k , and dividing by $(1-\rho_{k-1})$, we obtain:

$$W_k = \frac{1}{c\mu} \frac{\left\{ \frac{1}{1-\rho_k} - (M+1) \frac{\rho_k^M}{1-\rho_k^{M+1}} \right\}}{1-\rho_{k-1}} \cdot p(0 < W_k < \infty) \quad (1 < k \leq r) \quad (22)$$

Conclusions

We note that equations (9) and (10) are identical to those for infinite length queues [27, 53] and equations (11) and (12) differ only in the appearance of the term $(1-\rho_k^{M+1})$ which is just the effect of the finite length queue.

We also note, for finite queues, that since the arrival rate of priority k units drops to zero when the source is blocked, statistical equilibrium does not require that $\rho_k < 1$. Our only restriction was that $\rho_k \neq 1$, ($1 \leq k \leq r$). For the case in which $\rho_k = 1$ for some k , we have only to apply L'Hospital's rule, as required, to equations (11) through (22) possibly after some algebraic manipulation, to obtain the desired results.

For example, for ($1 < k \leq r$),

$$\hat{W}_k = \lim_{\rho_k \rightarrow 1} W_k = \frac{1}{c\mu} \left\{ \frac{1}{1-\rho_{k-1}} \frac{M+2}{2} \right\} \hat{p}(0 < W_k < \infty)$$

where

$$\hat{p}(0 < W_k < \infty) = \lim_{\rho_k \rightarrow 1} p(0 < W_k < \infty) = \left\{ \frac{M}{M+1} + \rho_{k-1}^M \frac{1-\rho_{k-1}}{1-\rho_{k-1}^{M+1}} \right\} p(>0)$$

We note that this procedure is necessary, at most, only for a single value of k , since $\rho_1 < \rho_2 < \dots < \rho_k < \dots < \rho$ and hence, at most, only one of the $\rho_k = 1$.

We conclude by arguing that equations (11) through (22) define a general multiserver model which, in addition to having as limiting cases the infinite length queue models [7, 8, 15, 27, 53] as well as the zero length queue model [47], is

also valid for all queue lengths between these two extremes and applies equally well to both nonpreemptive priority and first come, first served queue disciplines. As an example consider the case when M is infinite and $\rho_k < 1$. Then

$$\hat{W}_k = \lim_{M \rightarrow \infty} W_k = \frac{1/c\mu}{(1-\rho_k)(1-\rho_{k-1})} \lim_{M \rightarrow \infty} p(0 < W_k < \infty) = \frac{1/c\mu}{(1-\rho_k)(1-\rho_{k-1})} \hat{p}(> 0)$$

where

$$\hat{p}(> 0) = \lim_{M \rightarrow \infty} p(> 0) = \frac{1}{1-\rho} \frac{(c\rho)^c}{c!} p_0$$

which, except for notational differences, agrees with the well known results of Cobham [7, 8] and Holley [27]. (See also Saaty [53].) As a further example, consider the case for $r=1$, $c=1$, and $M=N-1$. Then

$$W_1 = \frac{1}{\mu} \left\{ \frac{1}{1-\rho} - (N) \frac{\rho^N}{1-\rho^N} \right\} p(> 0)$$

but for this case,

$$p(> 0) = \frac{\rho - \rho^{N+1}}{1 - \rho^{N+1}}$$

and so, after rearranging terms we have:

$$W_1 = \frac{\rho[1 - (N+1)\rho^N + N\rho^{N+1}]}{\mu(1-\rho^{N+1})(1-\rho)}$$

which agrees with the well known result of Saaty [55]. (See in particular equation 3-16 on page 61.)

Based on the essential characteristics of the various computers within the center we can employ the queueing theory model to facilitate studying the effects of priority assignment and job dispatching. We have devised [2] an algorithm for scheduling jobs in center when the jobs consist of dependent tasks having linear loss functions.

Dispatching

So far we have dealt with only the individual computing centers. Let us now turn our attention, briefly, to the entire network. During the normal course of processing jobs in a given center some of the resultant output is required at other centers. However, owing to the communities of interest that exist within each center, the intercommunication between centers is typically small. Earlier we noted that one computer in each center, designated the communications computer, was responsible for handling all such traffic. We may reasonably utilize the existing hardware and software facilities to increase the total reward, if possible. When the occasion presents itself it may be advantageous to transmit certain jobs from one center to another for processing and then subsequent transmission back of the results. Specifically, we may be able to improve the total network throughput by sending relatively high priority jobs from a heavily loaded center to one which is less heavily loaded.

We shall presume that we have a command network, one computing center in the network having been designated the command center. The command center periodically polls the other centers and depending on queue levels, requests arrival rate and processing time statistics. Then using the equations for W_r , the average waiting time by priority class, the command center can establish which centers, if any, should transmit jobs to other centers for processing and what fraction of the jobs in each priority class should be transmitted. We introduce the following definitions:

$W_{r,j}$ = the average waiting time of a priority r job at the j^{th} center.

$\lambda_{r,j}$ = the average arrival rate of priority r jobs at the j^{th} center.

t_{jk} = the transmission time between the j^{th} and k^{th} centers (including retransmission of results).

c_{jk} = the cost of transmitting a job from the j^{th} center to the k^{th} center (including retransmission of results).

b_i = the linear loss rate associated with the i^{th} job, U_i .

Now for a job originating at the j^{th} center, the average cost associated with processing the job at that center, $C_{r,j}$, is given by:

$$C_{r,j} = (W_{r,j} + \frac{1}{\mu}) b_i$$

The average cost associated with transmitting the job from the j^{th} center to the k^{th} center and processing it there, $C_{r,k}$, is given by:

$$C_{r,k} = (W_{r,k} + \frac{1}{\mu} + t_{jk}) b_i + c_{jk}$$

Then if $C_{r,k} < C_{r,j}$ it would be more economical to process the job at the k^{th} center. However, we must qualify this somewhat, since the presence of the job in the k^{th} center will delay all jobs of lower priority than r and all subsequent arrivals of priority r as well. Therefore we presume that throughout the network the jobs are assigned priorities according to monotonically nonincreasing b_i .

(We can use McNaughton's rule [44] here since the jobs are independent.)

Then, it follows directly that transmitting a priority r job, U_i , from center j to center k will result in a decrease in the system cost, if and only if,

$$\Delta W_r = W_{r,j} - W_{r,k} > t_{jk} + c_{jk}/b_i.$$

We have tacitly assumed here that the average processing time of the job is considerably larger than the handling time of the communications computer. Otherwise it would be more profitable to process the job on the communications computer rather than to transmit it. Also from the foregoing discussion we quickly conclude that the priority of a job should not be changed when it is transmitted between centers, even though we invest time and money to transmit the job. To give it a higher priority would delay jobs with greater loss rates and to lower its priority would advance jobs with lower loss rates. The time and money invested in the transmission is presumably less than the loss associated with leaving the job in the original center, else we would not have transmitted it. Thus once the job arrives at the new center it becomes indistinguishable from the other jobs of the same priority.

Next we turn to the problem of determining what fraction of the jobs in each priority class should be transmitted between centers. Even though the average waiting time by priority class is not a linear function of arrival rate, we can use a method of successive linear approximations to obtain a balance between the waiting times among the centers. Each time the command center polls the other centers the dispatching algorithm is used.

Dispatching Algorithm

We consider each priority level one at a time beginning with $r = 1$. For the r^{th} priority level:

- a) we let j denote the center having the largest value of W_r and k denote the center having the smallest value of W_r . Then we compute

$$\Delta W_r = W_{r,j} - W_{r,k}.$$
- b) if $\Delta W_r < t_{jk} + c_{jk}/b_i$, we say that the network is "balanced for priority r " and we continue with priority $r + 1$.

c) if $\Delta W_r > t_{jk} + c_{jk}/b_i$, we transmit a fraction $\Delta \lambda_r$ of the priority r jobs from center j to center k , where

$$\Delta \lambda_r = \frac{|\lambda_{r,j} - \lambda_{r,k}|}{2}$$

Note that we have taken the absolute magnitude of the difference in arrival rates in case $\lambda_{r,k} < \lambda_{r,j}$. Next we recompute the waiting times for priority r and lower jobs for centers k and j . We repeat the entire process until the network is balanced for priority r . Then we continue with priority $r + 1$. This difference averaging algorithm may be repeated until either the entire network is balanced or the available intercenter communications facilities are saturated.

Thus we see that the steady state equations for waiting times can be used on a successive approximation basis to perform dynamic load regulation in the network. As a result the command center can effectively utilize the inter-center communications facilities to improve the overall system throughput.

4. THE ROAD AHEAD

An interesting generalization of our model would be to consider the entire network from a dynamic point of view. We would begin by formulating a network model with a common, but distributed, queue of work served by K computing centers each having a different processing time rate, μ_k . Then given that jobs consist of dependent tasks having linear loss functions, we would attempt to formulate rules for dynamically dispatching tasks between centers taking into account not only the transmission times between centers and the cost of such transmissions, but also the time each task has spent in the center. A particularly intriguing approach here would be to use the b_i values to proportionally increase each task's priority as a function of its waiting time in the network.

As an alternative, we could use the expected processing cost as a characteristic. For example, at the University of Illinois computing center, as of May 1, 1970, with the implementation of the IBM 360 MVT systems, the dollar cost for Central Processor Unit (CPU) time on the 360/75 is computed as follows:

$$\text{Cost} = X_1(aX_2 + bX_3)(cX_4 + X_5)$$

Where

a = seconds of CPU time used

b = number of I/O requests

c = kilobytes of core memory used

and X_1 , X_2 , X_3 , X_4 , and X_5 are weighting factors, currently having the values 0.01, 4, 0.04, 0.0045, and 0.5, respectively.

If a user estimates a , b , and c when he submits a job, we may compute an expected cost. Then a particularly intriguing approach here would be to give jobs an initial priority according to monotonically nondecreasing expected cost and to subsequently increase each job's priority dynamically as a function of its waiting time in the center.

SELECTED BIBLIOGRAPHY

1. Bowdon, E. K., Sr., "Modeling and Analysis of a Network of Computers," Ph.D. dissertation, University of Iowa, Iowa City, Iowa, 1969.
2. Bowdon, E. K., Sr., "Priority Assignment in a Network of Computers," IEEE Transactions on Electronic Computers, November 1969.
3. Bowdon, E. K., Sr., "Network Computer Analysis," Proceedings of the 5th Hawaii International Conference on System Sciences, 1972, pp. 201-203.
4. Burke, P. J., "The Output of a Queueing System," Operations Research, Vol. 4 (1956), pp. 699-704.
5. Chang, W. and D. J. Wong, "Analysis of Real Time Multiprogramming," Journal of the ACM, Vol. 12 (1965), pp. 581-588.
6. Chang, W., "Queues with Feedback for Time-Sharing Computer System Analysis," Operations Research, Vol. 16 (1968), pp. 613-627.
7. Cobham, A., "Priority Assignment in Waiting Line Problems," Operations Research, Vol. 2 (1954), pp. 70-76.
8. Cobham, A., "Priority Assignment - A Correction," Operations Research, Vol. 3 (1955), p. 547.
9. Coffman, E. G., Jr. and R. C. Wood, "Interarrival Statistics for Time Sharing Systems," Communications of the ACM, Vol. 9 (1966), pp. 500-503.
10. Coffman, E. G., "Studying Multiprogramming Systems with the Queueing Theory," Datamation, Vol. 13 (1967), pp. 47, 49-51, 54.
11. Coffman, E. G., Jr., "Bounds on Parallel Processing of Queues with Multiple-Phase Jobs," Naval Research Logistics Quarterly, Vol. 14 (1967), pp. 345-366.
12. Conway, R. W., W. L. Maxwell, and L. W. Miller, Theory of Scheduling, Addison-Wesley Publishing Company, Reading, Massachusetts, 1967, (202 references).
13. Cox, D. R., and W. L. Smith, Queues, Methuen and Company, London, 1961.
14. Davis, R. H., "Waiting-Time Distribution of a Multiserver, Priority Queueing System," Operations Research, Vol. 14 (1965), pp. 133-136.
15. Dressin, S. A., and E. Reich, "Priority Assignment on a Waiting Line," Quarterly of Applied Mathematics, Vol. 15 (1957), pp. 208-211.
16. Eastman, W. L., S. Even, and I. M. Isaacs, "Bounds for the Optimal Scheduling of n Jobs on m Processor," Management Science, Vol. 11 (1964), pp. 268-279.
17. Eastman, W. L., "Comments on a Paper by Schild and Fredman," Management Science, Vol. 11 (1965), pp. 754-755.
18. Feller, W., An Introduction to Probability Theory and Its Applications, Vol. 1, John Wiley and Sons, New York, 1957.

19. Fife, D. W., "The Optimum Control of Queues with Applications to Computer Systems," Technical Report No. 170, Cooley Electronics Laboratory, University of Michigan, Ann Arbor, Michigan, 1965, 272 pages.
20. Fife, D. W., "Scheduling with Random Arrivals and Linear Loss Functions," Management Science, Vol. 11 (1965), pp. 429-437.
21. Fife, D. W., "Comment to the Editor," Management Science, Vol. 14 (1968), pp. 254-255.
22. Gaver, D. P., Jr., "Probability Models for Multiprogramming Computer Systems," Journal of the ACM, Vol. 14 (1967), pp. 423-438.
23. Graham, R. L., "Bounds for Certain Multiprocessing Anomalies," The Bell System Technical Journal, Vol. 45 (1966), pp. 1563-1581.
24. Greenberger, M., "The Priority Problem and Computer Time Sharing," Management Science, Vol. 12 (1966), p. 888.
25. Held, M., and R. M. Karp, "A Dynamic Programming Approach to Sequencing Problems," Journal of the Society of Industrial Applied Mathematics, Vol. 10 (1962), pp. 196-210.
26. Heyman, D. P., "Optimal Operating Policies for M/G/1 Queueing Systems," Operations Research, Vol. 16 (1968), pp. 362-382.
27. Holley, J., "Waiting Line Subject to Priorities," Operations Research, Vol. 2 (1954), pp. 341-343.
28. Jackson, J. R., "Networks of Waiting Lines," Operations Research, Vol. 5 (1957), pp. 518-521.
29. Jackson, J. R., "Some Problems in Queueing with Dynamic Priorities," Naval Research Logistics Quarterly, Vol. 7 (1960), pp. 235-249.
30. Jackson, J. R., "Queues with Dynamic Priority Discipline," Management Science, Vol. 8 (1961), pp. 18-34.
31. Jackson, R. R. P., and J. C. Henderson, "The Time Dependent Solution to the Many Server Poisson Queue," Operations Research, Vol. 14 (1966), pp. 720-723.
32. Jewell, W. S., "A Simple Proof of $L = \lambda W$," Operations Research, Vol. 15 (1967), pp. 1109-1106.
33. Joslin, E. O., "Application Benchmarks: The Key to Meaningful Computer Evaluations," Proceedings of the ACM 20th National Conference, August, 1965, pp. 27-37.
34. Joslin, E. O., and J. J. Aiken, "The Validity of Basing Computer Selections on Benchmark Results," Computers and Automation, Vol. 15 (1966), pp. 22-23.
35. Karlan, S., and J. McGregor, "Many Server Queueing Processes with Poisson Input and Exponential Service Times," Pacific Journal of Mathematics, Vol. 8 (1958), pp. 87-118.

36. Kiefer, J., and J. Wolfowitz, "On the Theory of Queues with Many Servers," American Mathematical Society Transactions, Vol. 78 (1955), pp. 1-18.
37. Kleinrock, L., "Theory of Queues Applied to Time-Shared Computer Systems," IEEE Region Six Conference Record, Vol. 2 (1966), pp. 491-500.
38. Kleinrock, L., "Time-Shared Systems: A Theoretical Treatment," Journal of the ACM, Vol. 14, (1967), pp. 242-261.
39. Kleinrock, L., and E. G. Coffman, "Distribution of Attained Service in Time-Shared Systems," Journal of Computer and System Sciences, Vol. 1 (1967), pp. 287-298.
40. Kleinrock, L., "Optimum Bribing for Queue Position," Operations Research, Vol. 15 (1967), pp. 304-318.
41. Krishnamoorthi, B., and R. C. Wood, "Time-Shared Computer Operations with Both Interarrival and Service Times Exponential," Journal of the ACM, Vol. 13 (1966), pp. 317-338.
42. Lawler, E. L., "On Scheduling Problems with Deferral Costs," Management Science, Vol. 11 (1964), pp. 280-288.
43. Little, J. D. C., "A Proof for the Queueing Formula $L = \lambda W$," Operations Research, Vol. 9 (1961), pp. 383-387.
44. McNaughton, R., "Scheduling with Deadlines and Loss Functions," Management Science, Vol. 6 (1959), pp. 1-12.
45. Manacher, G. K., "Production and Stabilization of Real-Time Task Schedules," Journal of the ACM, Vol. 14 (1967), pp. 439-465.
46. Miller, R. G., Jr., "Priority Queues," Annals of Mathematical Statistics, Vol. 31 (1960), pp. 86-103.
47. Morse, P. M., Queues, Inventories, and Maintenance, John Wiley and Sons, New York, 1958.
48. Richards, P., "Parallel Programming," Report No. TD-B60-27, Technical Operations, Inc., August, 1960, 26 pages.
49. Root, J. G., "Scheduling with Deadlines and Loss Functions on k Parallel Machines," Management Science, Vol. 11 (1965), pp. 460-475.
50. Rothkopf, M. H., "Scheduling Independent Tasks on One or More Processors," Interim Technical Report No. 2, MIT Operations Research Center, January, 1964, 111 pages.
51. Rothkopf, M. H., "Scheduling Independent Tasks on Parallel Processors," Management Science, Vol. 12 (1966), pp. 437-447.
52. Rothkopf, M. H., "Scheduling with Random Service Times," Management Science, Vol. 12 (1966), pp. 707-713.
53. Saaty, T. L., "Resume of Useful Formulas in Queueing Theory," Operations Research, Vol. 5 (1957), pp. 161-200, (68 references).

54. Saaty, T. L., "Time Dependent Solution of the Many Server Poisson Queue," Operations Research, Vol. 8 (1960), pp. 755-772.
55. Saaty, T. L., On Elements of Queueing Theory with Applications, McGraw-Hill, New York, 1961, (910 references).
56. Saaty, T. L., "Seven More Years of Queues - A Lament and a Bibliography," Naval Research Logistics Quarterly, Vol. 13 (1966), pp. 447-476, (594 references).
57. Scherr, A. L., "An Analysis of Time-Shared Computer Systems," Research Monograph No. 36, The MIT Press, Cambridge, Massachusetts, 1967, 115 pages.
58. Schild, A., and I. J. Fredman, "On Scheduling Tasks with Associated Linear Loss Functions," Management Science, Vol. 7 (1961), pp. 280-285.
59. Schrage, L. E., and L. W. Miller, "The Queue M/G/1 with the Shortest Remaining Processing Time Discipline," Operations Research, Vol. 14 (1966), pp. 670-683.
60. Schrage, L. E., "A Proof of the Optimality of the Shortest Remaining Processing Time Discipline," Operations Research, Vol. 16 (1968), pp. 687-690.
61. Shemer, J. E., "Some Mathematical Considerations of Time-Sharing Scheduling Algorithms," Journal of the ACM, Vol. 14 (1967), pp. 262-272.
62. Shemer, J. E., and G. A. Shippey, "Statistical Analysis of Paged and Segmented Computer Systems," IEEE Transactions on Electronic Computers, Vol. EC-15 (1966), pp. 855-863.
63. Takacs, L., "Priority Queues," Operations Research, Vol. 12 (1964), pp. 63-74.
64. Votaw, D. G., Jr., and A. M. Stover, "Queueing Systems with Bounded Waiting-Line Length," Operations Research, Vol. 14 (1965), pp. 317-328.

BIBLIOGRAPHIC DATA SHEET		1. Report No. UIUCDCS-R72-505	2.	3. Recipient's Accession No.	
4. Title and Subtitle Network Computer Analysis				5. Report Date January 1972	
				6.	
7. Author(s) Edward K. Bowdon, Sr.				8. Performing Organization Rept. No.	
9. Performing Organization Name and Address Department of Computer Science University of Illinois Urbana, Illinois 61801				10. Project/Task/Work Unit No.	
				11. Contract/Grant No. NSF GJ 28289	
12. Sponsoring Organization Name and Address National Science Foundation Washington, DC				13. Type of Report & Period Covered Research	
				14.	
15. Supplementary Notes Presented at the 5th Hawaii International Conference on System Sciences					
16. Abstracts (see reversed side of this form)					
17. Key Words and Document Analysis. 17a. Descriptors Network Computer, Queueing Theory, System Modeling, Throughput Analysis, Dispatching, Load Leveling					
17b. Identifiers/Open-Ended Terms					
17c. COSATI Field/Group					
18. Availability Statement Release unlimited			19. Security Class (This Report) UNCLASSIFIED		21. No. of Pages 28
			20. Security Class (This Page) UNCLASSIFIED		22. Price

ABSTRACT

This research is aimed at developing analytical tools for system modeling and analysis of real time computer networks. We begin by formulating a geographically distributed network of computers.

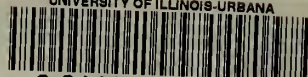
From a postulation of the essential characteristics of our computer network, we develop a queueing theory model for a multiserver system with a finite length priority queue. Then, under the assumptions of Poisson input and exponentially distributed processing times, we utilize this idealized mathematical model to investigate the steady state stochastic behavior of jobs in the network. We are particularly interested in the efficiency of computer utilization and the average waiting time for jobs of different priority classes.

We conclude by employing the average waiting time results from the queueing theory model to determine steady state job dispatching rules for a network of centers. From a knowledge of the average arrival rates and processing times for each priority class, we determine what fraction of the jobs in each class should be transmitted between centers in order to balance the average waiting times for each priority class throughout the network.

JUN 7 1972



UNIVERSITY OF ILLINOIS-URBANA



3 0112 039462434